ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/221123822

Color printer characterization in MATLAB

Conference Paper *in* Proceedings / ICIP ... International Conference on Image Processing · January 2002

DOI: 10.1109/ICIP.2002.1038059 · Source: DBLP

CITATIONS
4

READS
120

2 authors, including:

Henry Trussell
North Carolina State University
208 PUBLICATIONS **5,181** CITATIONS

SEE PROFILE

All content following this page was uploaded by Henry Trussell on 03 April 2020.

The user has requested enhancement of the downloaded file.

Color Printer Characterization in MATLAB

M. J. Vrhel
Color Savvy Systems Ltd.
724 Pleasant Valley Drive
Springboro, OH 45066

H. J. Trussell
Dept. of Electrical and Computer Engineering
North Carolina State University
Raleigh, NC 27695-7911

Abstract

Many researchers are working on various aspects of color imaging. However, it is apparent from the papers published that it is difficult for many of them to produce high quality color reproductions. While our previous work outlined the steps needed to produce calibrated color images, the problem of creating the programs to implement the process locally seems to be a serious problem. To surmount this final barrier, we have developed a web site that has the MATLAB programs necessary to characterize printers and produce high quality images.

1 Introduction

Our previous work [1], provided a high level mathematical description of the problem of producing accurate color. There appear to be two significant difficulties for many researchers: 1) the cost in time and energy to implement the software solutions described in [1], and 2) the access to a color measurement device, e.g., colorimeter or spectrophotometer. The goal of this work is to provide MATLAB routines that solve the first problem. Users must still obtain a measurement device if accurate color is desired. Researchers can use this software to characterize their color printers, and hopefully improve the quality of their color printing. We will first describe the problem mathematically, provide one solution, and end with an example.

2 The Problem

The printer characterization problem is to determine a mapping that takes a CIE colorimetric value and finds the "best" printer control value (RGB value) to create that color. Similar methods can be used for use with softcopy. Ideally, the mapping is determined as follows:

Let $\mathcal{F}$ be the mapping from printer control values (RGB values) to CIEL*a*b* values, and assume that this mapping is known. The problem, is to determine a mapping $\mathcal{G}_{optim}$, from CIEL*a*b* to RGB, such that

$$\mathcal{G}_{optim} = \arg(\min_{\mathcal{G}} E\{||\mathcal{F}(\mathcal{G}(\mathbf{t})) - \mathbf{t}||\}) \qquad (1)$$

where $\mathbf{t}$ is a CIEL*a*b* value, and the expectation value is taken over these colorimetric values. Often, the mapping $\mathcal{F}$ is referred to as the forward mapping and $\mathcal{G}$ is referred to as the inverse mapping

3 A Solution

The methods for characterization can be divided into parametric and non-parametric based approaches. The parametric based methods require the estimation of model parameters using data collected from a colorimeter. The non-parametric based methods require the estimation of look-up-table (LUT) entries or possibly neural net coefficients.

Printer characterization is difficult due to the inherent nonlinearity of the printing process, and the wide variety of methods used for color printing (e.g. lithography, inkjet, dye sublimation, etc.). Because of these difficulties, printing devices are often profiled with a multi-dimension LUT (MLUT) and interpolation [2, 3, 4]. In other words, the continuum of values is found by interpolating between points in the MLUT. For particular printing methods, model based methods are also used for characterizing printers [5, 6].

In this paper, we will describe an MLUT approach to printer characterization. The approach is as follows:

- Print an $N \times N \times N$ RGB sequence on the printer. The term $N \times N \times N$ RGB sequence denotes a sequence of N^3 values that give a good coverage of the 3-D printer RGB color space. As an example, a $3 \times 3 \times 3$ RGB sequence would be given by (from left to right and then top to bottom – black to

white)

0, 0, 0	0, 0, 127	0, 0, 255
0, 127, 0	0, 127, 127	0, 127, 255
0, 255, 0	0, 255, 127	0, 255, 255
127, 0, 0	127, 0, 127	127, 0, 255
127, 127, 0	127, 127, 127	127, 127, 255
127, 255, 0	127, 255, 127	127, 255, 255
255, 0, 0	255, 0, 127	255, 0, 255
255, 127, 0	255, 127, 127	255, 127, 255
255, 255, 0	255, 255, 127	255, 255, 255

- Measure each of the N^3 printed samples with a colorimeter and obtain the CIEL*a*b* values.
- Approximate the forward mapping $\mathcal{F}$ by the construction of an $N \times N \times N$ MLUT.
- Build an $M \times M \times M$ MLUT in which each grid point is obtained by solving Equation(1), thereby *providing an approximation to the inverse mapping* $\mathcal{G}$.

The most time-consuming part of the process is the measurement and recording of the sample values. The only computationally difficult problem in the above steps is the construction of the mapping $\mathcal{G}$ (the CIEL*a*b* to RGB mapping). To simplify the description of our construction process, we will use the following notation:

- Given an $N \times N \times N$ RGB to CIEL*a*b* MLUT $\mathcal{F}$, let the CIEL*a*b* value in the table at index $[i, j, k]$ be denoted by

$$F_N([i, j, k]) \quad i, j, k \in [0, 1, ..., N-1] \qquad (2)$$

Note that F_N is a 3-vector, representing L^*, a^* and b^* values.

- Define a cubic collection of table indices as

$$\begin{aligned} C_{[i,j,k]} = \quad & \{F_N([i, j, k]), F_N([i+1, j, k]), \\ & F_N([i, j+1, k]), F_N([i, j, k+1]), \\ & F_N([i+1, j+1, k]), F_N([i+1, j, k+1]), \\ & F_N([i, j+1, k+1]), F_N([i+1, j+1, k+1])\} \end{aligned}$$

- Given a cubic collection of table indices $C_{[i,j,k]}$ and an RGB value $\mathbf{v}$ that is within that cube, the MLUT interpolated CIEL*a*b* value will be determined by the function

$$\mathcal{H}(\mathbf{v}, C_{i,j,k}, [i, j, k], N, \text{RANGE}(\mathbf{v})) \qquad (3)$$

where RANGE($\mathbf{v}$) returns the minimum and maximum values possible for $\mathbf{v}$, and the interpolation function can be any desired multi-dimension interpolation function (e.g. trilinear, tetrahedral, pyramid, etc) [3].

Using the above notation, the construction of the inverse MLUT table $\mathcal{G}$ is performed as follows:

- Take an arbitrary grid point $\mathbf{p}$ (a CIEL*a*b* value) of the (M^3) points that we wish to fill. The goal is to determine what RGB value should be associated with that CIEL*a*b* grid point.
- Of the N^3 CIEL*a*b* values that were measured, find the closest one to $\mathbf{p}$ in terms of Euclidean distance (ΔE_{a*b*} distance), i.e. find

$$[I, J, K]^T = \arg(\min_{i,j,k} ||F_N([i, j, k]^T) - \mathbf{p}||^2) \qquad (4)$$

where $i, j, k \in [0, 1, ..., (N-1)]$

- For the indices $[I, J, K]$ determine the eight (or fewer if an edge point) cubes that surround the point in the $N \times N \times N$ notation of the data. Assuming $1 < I, J, K < N-2$, these eight cubes would consist of the cubes $C_{[I-1,J-1,K-1]}$, $C_{[I-1,J-1,K]}$, $C_{[I-1,J,K-1]}$, $C_{[I,J-1,K-1]}$, $C_{[I,J,K-1]}$, $C_{[I,J-1,K]}$, $C_{[I-1,J,K]}$, $C_{[I,J,K]}$. For ease of notation, denote these eight cubes simply as C_l $l = 1, ..., 8$.
- For each cube, solve the following optimization problem

$$\mathbf{v}_{optim,l} = \arg(\min_{\mathbf{v}} ||\mathbf{p} - \mathcal{H}(\mathbf{v}, C_l, [i, j, k], N, \text{RANGE}(\mathbf{v}))||^2) \qquad (5)$$

using a nonlinear programming method.

- If the RGB vector $\mathbf{v}_{optim,l}$ is within the cube C_l[1], then stop the optimizations and fill the grid point $\mathbf{p}$ that we are trying to fill with the RGB value $\mathbf{v}_{optim,l}$.
- If the RGB vectors $\mathbf{v}_{optim,l}$ are all outside of their respective cubes, then the CIEL*a*b* value $\mathbf{p}$ is outside of the printer gamut (in other words the printer cannot create that color). In this case, the following optimization problems are solved

$$\delta e(l) = (\min_{\mathbf{v}} ||\mathbf{p} - \mathcal{H}(\mathbf{v}, C_l, [i, j, k], N, \text{RANGE}(\mathbf{v}))||^2)$$

where $l = 1, ..., 8$ and with the constraint that the RGB vector $\mathbf{v}$ must remain in the cube C_l. The optimal constrained arguments $\mathbf{v}_{outgamut,l}$ are saved with each computation. Once the $\delta e(l)$ values are determined and the minimum error

$$L = \arg(\min_{l} \delta e(l))$$

[1]This test is easily done by comparing $\mathbf{v}_{optim,l}$ to the RGB grid points in C_l

is found, the gamut mapped RGB vector $\mathbf{v}_{outgamut,L}$ is placed in the grid point $\mathbf{p}$.

Note that many different approaches to gamut mapping could be used in the above algorithm.

4 The Subroutines

The MATLAB code to implement the above operations is contained at NCSU FTP site: ftp.eos.ncsu.edu. The directory is pub/hjt/icip2002. It can be accessed via browser at ftp://ftp.eos.ncsu.edu/pub/hjt/icip2002/ The primary routines are as follows:

ChartImage=MakeChart(N,SquareSize): Where *ChartImage* is an image of the $N \times N \times N$ RGB sequence to be printed and then measured with the colorimeter. The variable *SquareSize* sets the dimension of each square to be measured in terms of number of pixels. The sampling step along each axis in the 3 dimensional RGB space is computed as $k * 255/(N-1)$ where $k = 0, 1, ..., N-1$. Note that the samples should be measured left to right and top to bottom (black to white) as the image is created.

F=Buildrgb2lab(labs,N): Where *labs* is a vector of the N^3 chart samples that were measured with a colorimeter, N is the size of the table that you wish to generate, and F is the RGB to CIEL*a*b* MLUT for the printer. This function performs no mathematical operations on the measured samples. The samples are simply reordered into a MATLAB multidimensional array of size $3 \times N \times N \times N$.

G=Buildlab2rgb(F,M,N): Where M is the desired size of the table you wish to generate, and G is the CIEL*a*b* MLUT for the printer. This routine solves the cost functions in Equations (4) and (5) using trilinear interpolation for the function $\mathcal{H}$.

printimage=MLUTinterp(labimage,G,M): Where *labimage* is the input image in CIEL*a*b* space and *printimage* is the image that should be printed on the printer. This function uses trilinear interpolation to apply the MLUT to the input image *labimage*.

Since the above routine uses a CIEL*a*b* image as input, the user must either have a CIEL*a*b* image or convert an image to CIEL*a*b*. Because CIEL*a*b* images are currently rare, we include a routine to convert an RGB image to CIEL*a*b*. Note that some image processing software, such as, Photoshop, can convert files to CIEL*a*b*. To transform an RGB image to CIEL*a*b*, it is necessary to know what the given RGB values mean colorimetrically. Since sRGB is becoming more common [7], we also provide a function that transforms from sRGB to CIEL*a*b*.

labimage=Srgb2lab(sRGBimage): Where *sRGBimage* is the input image.

To provide some speed-up , the web site contains C-MEX files for some of the above routines. In those cases, the routine name has a prefix C added (e.g. *CMLUTinterp* for *MLUTinterp*)

5 An Example

As an example of using the above routines, we characterized an EPSON 5000 color ink-jet printer. The process was performed as follows:

- A $9 \times 9 \times 9$ RGB chart was created using the *MakeChart* function. This chart was then printed on the EPSON 5000 printer.
- A Gretag SpectroScan instrument was used to obtain the 729 CIEL*a*b* values. Any reflectance colorimeter can be used, however the characterization is going to be valid only for the illuminant associated with the white point of the CIEL*a* b* that is used for the recorded values.
- The 729 values were read into MATLAB as the variable *labs*.
- The function *F=Buildrgb2lab(labs,9)* was executed.
- The function *F=Buildlab2rgb(F,31,9)* was executed. Depending upon your processing power this function takes a long time at a table size of M=31. However, this function will need to be run only once to characterize the printer for as long as it printing is stable. If the printer is subject to "drift", the characterization will have to be done as often as necessary.
- An sRGB version of the Lena image (scanned by the authors and available at the web site) was read into MATLAB using MATLAB's command *sRGBimage=imread('LenaSrgb.tiff')*.
- The sRGB image was converted to CIEL*a* b* using the command *labimage=CSrgb2lab(sRGBimage)*.
- The CIEL*a*b* image *labimage* was converted to the printer RGB space using the MLUT interpolation function *printimage=CMLUTinterp(labimage,G,31)*.
- The image *printimage* was printed on the printer providing a good match to the original Lena image.

Figure 1: sRGB version of Lena

Figure 2: Printer RGB version of Lena

The sRGB version of the Lena image is shown in Figure 1. If the reader is viewing this paper with an sRGB calibrated monitor or printed a copy on an sRGB calibrated printer, then this image represents how the Lena image should appear when properly displayed. Figure 2 shows the RGB image *printimage*, which when printed on our Epson 5000 printer will create the Lena image as it should properly appear. Note that the two images do not match, which is an indication of the magnitude of error that would occur if the sRGB image was printed directly on the Epson 5000 printer. Using the above routines to create and apply the transformation $\mathcal{G}$ to the sRGB image enabled us to obtain the proper image from our printer.

6 Conclusion

The goal of this work was to provide researchers with a tool to enable accurate printing of color images. It is hoped that researchers will also take these routines and improve upon them in terms of their accuracy and efficiency.

References

[1] M. J. Vrhel and H. J. Trussell, "Color Device Calibration: A Mathematical Formulation," *IEEE Transactions on Image Processing*, Vol. 8, No. 12, Dec. 1999.

[2] M. C. Stone, W. B. Cowan, and J. C. Beatty, "Color Gamut Mapping and the Printing of Digital Color Images," *ACM Transactions on Graphics*, Vol. 7, No. 4, Oct. 1988.

[3] H. R. Kang, Color Technology for electronic devices, SPIE Press, Bellingham WA, 1997.

[4] R. Balasubramanian, "Reducing the cost of lookup table based color transformations," *Proc. IS&T/SID Seventh Color Imaging Conference: Color Science, Systems and Applications*, pp. 65-68, November, 1999.

[5] M. Xia, E. Saber, G. Sharma, and A. M. Tekalp, "End-to-End Color Printer Calibration by Total Least Squares Regression," *IEEE Trans. on Image Proc.*, vol. 8, no. 5, pp. 700-716, May 1999.

[6] R. Balasubramanian, "Colorimetric Modeling of Binary Color Printers," *Proc. IEEE Intl. Conference on Image Proc.,*, Vol. 2, pp. 327-330, Nov. 1995.

[7] M. Anderson *et. al.* , " Proposal for a Standard Default Color Space for the Internet," *IS&T/SID 4th Color Imaging Conference*, pp. 238-246, 1996.